Downloaded from orbit.dtu.dk on: May 05, 2023

DTU Library

Generalization: The Hidden Agenda of Learning.

The Past, Present and Future of Neural Networks for Signal

Larsen, Jan; Hansen, Lars Kai

Published in:
I E E E - Signal Processing Magazine

Link to article, DOI:
10.1109/MSP.1997.637310

Publication date:
1997

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Larsen, J., & Hansen, L. K. (1997). Generalization: The Hidden Agenda of Learning. The Past, Present and Future of Neural Networks for Signal. *I E E E - Signal Processing Magazine*, *14*(6), 43-45. https://doi.org/10.1109/MSP.1997.637310

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

practical algorithms. One such application is an intelligent user interface agent where a software program is able to adjust its behavior based on observation of users' action as well as interaction with the user through some sort of dialog. For example, in modern speech-recognition software, speaker adaptation is an important component. The current approach is to implement the speaker-adaptation unit as a passive learner that will learn new information about the speaker whenever the speaker points out the mistake in the recognition results. It is possible that the user must perform numerous corrections before the speaker adaptation is fully adapted to the speaker-specific acoustic characteristics. On the other hand, an active speaker-adaptation algorithm would be able to request the speaker to read some specific sentences and quickly adapt to speaker's characteristics. Another example is an intelligent information browser [10] where the browser is able to aid a novice user in locating the most relevant documents by pre-classifying the documents using clustering. Then it would ask user to clarify the ambiguity by presenting the user with sample documents that locate near the present estimate of the decision boundary. An initial prototype of this tool has shown great potential [10].

References

1. M.D. Richard and R. P. Lippmann, "Neural Network Classifiers Estimate Bayesian A Posterior Probabilities," *Neural Computation*, 3(4):461-483, 1991.
2. B. Ripley, *Stochastic Simulation*. Wiley, New York, 1987.
3. David A. Cohn, "Neural Network Exploration Using Optimal Experiment Design," in *Advances in Neural Information Processing Systems 6*, 1994.
4. Mark Plutowski and Halbert White, "Active Selection of Training Examples for Network Learning in Noiseless Environments," Tech. Report CS91-180, Univ. of California-San Diego, Feb. 1990.
5. Jenq-Neng Hwang, Jai J. Choi, Seho Oh, and Robert J. Marks II, "Query-Based Learning Applied To Partially Trained Multilayer Perceptrons," *IEEE Transactions on Neural Networks*, vol. 2, no. 1, pp. 131-136, Jan. 1991.
6. Yoshiyuki Kabashima and Shigeru Shinomoto, "Incremental Learning with and without Queries in Binary Choice Problems," in *Proc. International Joint Conference on Neural Networks*, 1993, vol. 2, pp. 1637-1640.
7. Jong-Min Park, "On-Line Learning using Pattern Recognition and Active Learning," *Proc. ICASSP'97*, Munich, Germany, pp. 3217-3220.
8. T. Kohonen, "Improved Version of Learning Vector Quantization," International Joint Conf. on Neural Networks, Vol. I, pp. 545-550, San Diego, June 1990.
9. Jong-Min Park and Yu Hen Hu, "On-Line Learning for Active Pattern Recognition," *IEEE Signal Processing Letters*, November 1996.
10. Jong-Min Park, "Intelligent Information Query and Browsing System Using Active Learning," Ph.D. Dissertation, Dept. of Electrical and Computer Engineering, University of Wisconsin, August 1997.

Generalization: The Hidden Agenda of Learning

Jan Larsen and Lars Kai Hansen,
Technical University of Denmark

Most neural systems are adapted by optimization of a performance index, typically the minimization of a "cost function," based on a finite database (a training set) of N noisy examples derived from the target system. However, there is always the ***hidden agenda*** that the model should perform well, not only on the training set, but on the much larger set of future inputs to the system.

While reading for your finals you solve the previous years' tests, but you know very well that if you then test yourself on last year's test, the result will be biased—too optimistic! Only a test on a fresh data set, a test that was put aside before you started reading, will give the you a reliable prediction of the final performance.

Doing well on unseen data may at first seem unattainable, but the ability to generalize in very complex environments is nevertheless one of the most striking properties of neural systems, and indeed one of the reasons that neural networks have shown useful in practical applications.

As an example: in [1], a neural-network system for inspection of handwritten digits was able to classify 99.98% correct after training on a data base of 7291 digits and classify 95 % correct on an additional test set of 2007 digits.

When using a super-flexible model family, like neural networks, which in principle can model arbitrarily complex systems, *overfit* is a major concern, which finds expression in the ubiquitous bias-variance dilemma [2]. The generalization ability of an adaptive system is the quantitative measure of performance on a hypothetical infinite test set. While this quantity cannot be accessed directly, algebraic asymptotic estimates of generalization, valid for large training sets ($N \rightarrow \infty$), can be derived [3-8]. Such asymptotic results were earlier derived for supervised learning; however, it was recently shown that generalization ability for unsupervised learning machines (e.g., principal component analysis and clustering schemes) can be analyzed in a similar framework [9].

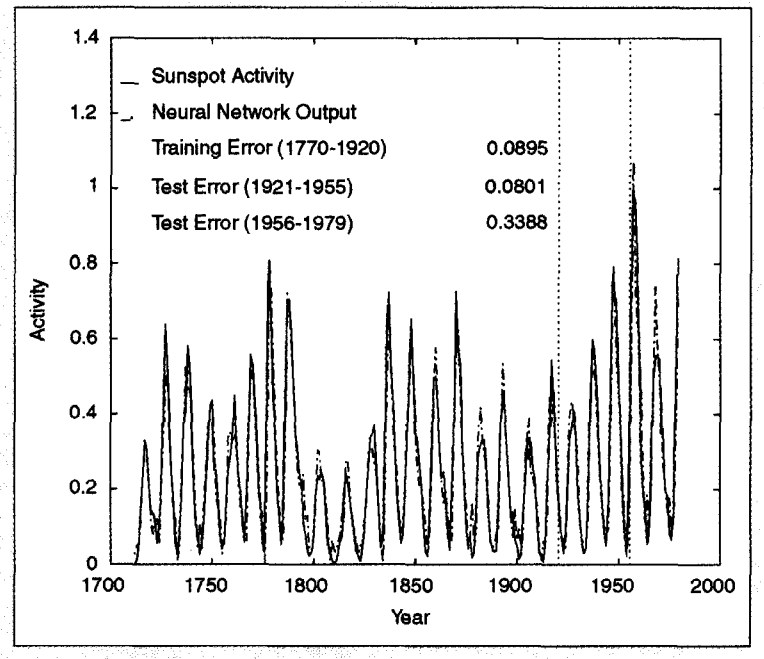
If sufficient computational capacity is available, empirical resampling schemes can be invoked. The two basic resampling strategies are cross-validation and bootstrap. Cross-validation [10, 11] is based on a random division of the database into disjunct training and validation sets. The procedure can be repeated, leading to more accurate results at the price of increased computation. The so-called leave-one-out cross-validation is based on using only a single example in the test set, and typically resampling N times. Approximative techniques, by which the computational overhead in leave-one-out is significantly reduced, have been reported [7, 12].

Bootstrap, invented by Efron [13], is based on resampling with replacement. Bootstrap produces pseudo training sets of size N, and, hence, simulates training set fluctuations at the full sample size. It has been applied to control overfit in a number of investigations [14-16].

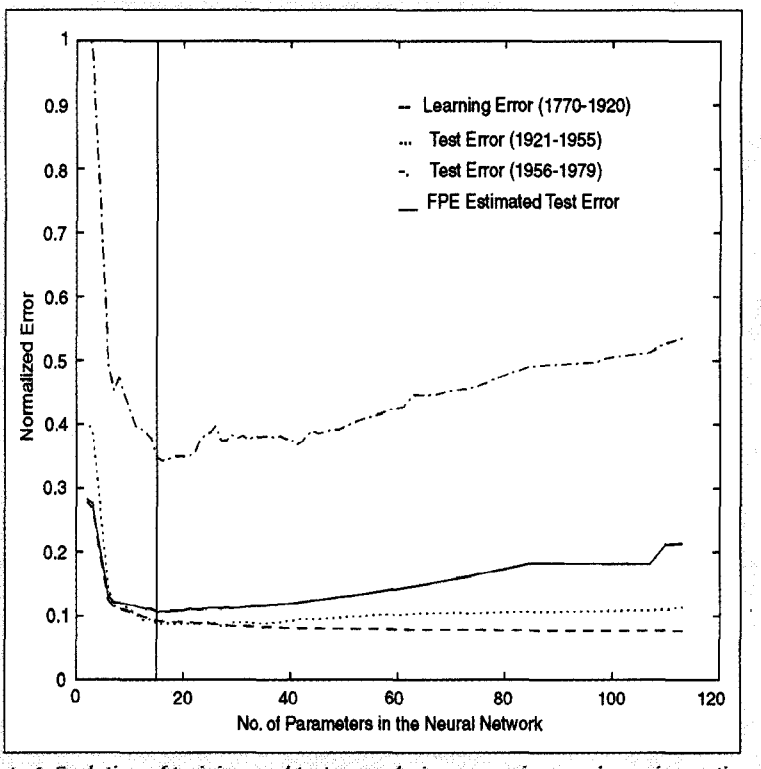
Optimization of the neural-network architecture may lead to better generalization ability and preferably lower computational burden. Optimizing the network architecture is to optimally trade off bias and variance [2], hence, maximizing generalization ability. This can be done *di-*

Authorized licensed use limited to: Danmarks Tekniske Informationscenter. Downloaded on April 13,2010 at 12:14:56 UTC from IEEE Xplore. Restrictions apply.

rectly by optimizing the structure of the network by pruning or growing techniques or *indirectly* by using regularization. Regularization, which goes back to Hadamard, consists of adding a penalty term to the cost function. As an example, consider predicting the sunspot time series shown in Fig. 3. In contrast, Fig. 4 [17] shows that generalization error (test error) is reduced by pruning the network.



▲ *3. Prediction of sunspot time series using an optimally pruned feed-forward neural network.*



▲ *4. Evolution of training and test error during a pruning session using optimal brain damage [18]. FPE is a modified version the final prediction error estimate [3]. The vertical lines indicate the optimal network for which the FPE estimate is minimal.*

References

1. Y. Le Cun, et al., "Back-propagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, vol. 1, no. 4, pp. 541-551, 1989.
2. S. Geman, E. Bienenstock & R. Doursat, "Neural Networks and the Bias/Variance Dilemma." *Neural Computation*, vol. 4, pp. 1-59, 1992.
3. H. Akaike, "Fitting Autoregressive Models for Prediction," *Ann. Inst. Stat. Mat.*, vol. 21, pp. 243-247, 1969.
4. P. Craven & G. Wahba, "Smoothing Noisy Data with Spline Functions: Estimating the Correct Degree of Smoothing by the Method of Generalized Cross-Validation," *Numerical Mathematics*, vol. 31, 377-403, 1979.
5. J. Larsen, "A Generalization Error Estimate for Nonlinear Systems," in S.Y. Kung, F. Fallside, J. Aa. Sorensen & C.A. Kamm (eds.) *Neural Networks for Signal Processing 2: Proceedings of the 1992 IEEE-SP Workshop*, Piscataway, New Jersey: IEEE, 1992, pp. 29-38.
6. J. Moody, "Note on Generalization, Regularization, and Architecture Selection in Nonlinear Learning Systems," in B.H. Juang, S.Y. Kung & C.A. Kamm (eds.) *Proceedings of the First IEEE Workshop on Neural Networks for Signal Processing*, Piscataway, New Jersey: IEEE, pp. 1-10, 1991.
7. J. Moody, "Prediction Risk and Architecture Selection for Neural Networks" in V. Cherkassky, J.H. Friedman & H. Wechsler (eds.) *From Statistics to Neural Networks: Theory and Pattern Recognition Applications*, Series F, vol. 136, Berlin, Germany: Springer-Verlag, 1994.
8. N. Murata, S. Yoshizawa and & S. Amari, "Network Information Criterion - Determining the Number of Hidden Units for an Artificial Neural Network Model," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 865-872, Nov. 1994.
9. L.K. Hansen & J. Larsen, "Unsupervised Learning and Generalization," *Proceedings of IEEE International Conference on Neural Networks*, Washington DC, vol. 1, pp. 25-30, June 1996.
10. S. Geisser, "The Predictive Sample Reuse Method with Application," *Journal of The American Statistical Association*, vol. 50, pp. 320-328, 1975.
11. M. Stone, "Cross-validatory Choice and Assessment of Statistical Predictors," *Journal of the Royal Statistical Society B*, vol. 36, no. 2, pp. 111-147, 1974.
12. L.K. Hansen & J. Larsen, "Linear Unlearning for Cross-Validation," *Advances in Computational Mathematics*, vol. 5, pp. 269-280, 1996.
13. B. Efron & R. Tibshirani, *An Introduction to the Bootstrap*, New York, NY: Chapman & Hall, 1993.
14. B. Efron & R. Tibshirani, "Cross-Validation and the Bootstrap: Estimating the Error Rate of a Prediction Rule," Techn. Report no. 477, Dept. of Statistics, Stanford University, May 1995. To appear in *Journ. Amer. Statist.*
15. R. Tibshirani, "A Comparison of Some Error Estimates for Neural Network Models," *Neural Computation*, vol. 8, pp. 152-163, 1996.
16. A.S. Weigend & B. LeBaron, "Evaluating Neural Network Predictors by Bootstrapping," in *Proceedings of the International Conference on Neural Information Processing* (ICONIP'94)}, Seoul, Korea, pp. 1207-1212, 1994.

Authorized licensed use limited to: Danmarks Tekniske Informationscenter. Downloaded on April 13,2010 at 12:14:56 UTC from IEEE Xplore. Restrictions apply.

17. C. Svarer, L.K. Hansen & J. Larsen, "On Design and Evaluation of Tapped-Delay Neural Architectures," in *Proceedings of the IEEE International Conference on Neural Networks*, San Francisco, California, USA, vol. 1, pp. 46-51, 1993.

18. Y. Le Cun, J.S. Denker & S.A. Solla, Optimal Brain Damage. In D.S. Touretzky (ed.), *Advances in Neural Information Processing Systems 2, Proceedings of the 1989 Conference*, San Mateo, California: Morgan Kaufmann Publishers pp. 598-605, 1990.

On-Line Step-Size Selection for Training of Adaptive Systems

Scott C. Douglas, University of Utah, and
Andrzej Cichocki, RIEKN

An adaptive system is by its very nature time-varying. The rate at which such a system changes its internal parameters determines its capabilities to adjust to and to obtain useful information from an unknown physical environment. One of the simplest and most-popular techniques for adjusting an adaptive system's parameters is the *gradient-descent* method, in which the parameters are changed according to the derivatives of a particular cost function with respect to the current parameter values. Both the least-mean-square (LMS) algorithm for adaptive filters [1] and the back-propagation algorithm for multilayer perceptrons [2] are gradient-descent methods. Much is known about the behavioral characteristics of gradient adaptation, and the algorithms are usually numerically robust. In gradient descent, the step sizes control the magnitudes of the changes in the parameters in the negative direction of the gradient.

The Kalman filter forms the basis for another class of parameter-estimation techniques that employ second-order information about the cost function being minimized. Recursive least-squares (RLS) techniques are widely used in linear estimation tasks and [3] explore the connections between RLS and Kalman techniques. The extended Kalman filter is a linearized version of the Kalman filter for nonlinear state-space estimation tasks [4], and it has been successfully applied to multilayer perceptron training [5]. Many simplified and approximate versions of this algorithm have been studied. Note that in RLS and linearized least-squares methods with forgetting factor λ, the parameter $(1 - \lambda)$ plays the role of the step size, whereas in Kalman techniques the step size is automatically determined from its underlying Bayesian problem formulation.

Goal of Step-Size Selection

The performance of any adaptive system that is attempting to drive its adjustable parameters to an optimum fixed set of parameters is governed by two quantities: (i) its *convergence rate* and (ii) the *misadjustment* in steady state. The convergence rate refers to the transient behavior of the parameters as they approach their optimum values. Misadjustment refers to the additional error in the output of the system caused by the random fluctuations of the parameters in steady-state. Generally speaking, the convergence rate of a system increases for step sizes that are somewhat less than one-half of the maximum value of the step size that provides stable adaptive behavior. In contrast, the misadjustment generally decreases as the step size is decreased.

The goal of any time-varying step size procedure is to increase the step size to a large but stable operating value when the parameters are some distance from their optimum settings and to systematically decrease the step size to reduce the misadjustment when the parameters are in the vicinity of their optimum settings. Since stability is often not explicitly ensured within a time-varying step-size method, it is generally necessary to limit the range of step sizes to guarantee stable operation of the system.

When the desired parameter settings vary with time, the system must continually readjust its parameters to follow these variations. The error induced in the model parameters by any time-variation of the unknown optimal parameters is called the *lag error*. In some cases where the velocity of the model parameters is constant, an optimum step-size value exists that minimizes the contributions of the misadjustment and the lag error after all initial transients in the parameters have died out.

We can classify a step-size selection method as *adaptive* or *nonadaptive* depending on its form. Nonadaptive methods calculate a time-varying step size according to *a priori* knowledge about the signals being processed, the cost function being optimized, and/or the parameter structure of the system. These techniques include asymptotically optimal methods as derived via the theory of stochastic-approximation [6] methods based on a statistical analysis of the particular system [7, 8] and heuristic approximations to these methods, commonly known as "search-and-converge" [9] or "gearshifting." By contrast, adaptive methods are based on on-line measurements of the state of the adaptive system, usually as characterized by the outputs or by the parameter updates of the system.

Nonadaptive step-size methods usually require more information about the adaptive system and the problem context than do adaptive step-size methods. However, nonadaptive step-size methods usually outperform adaptive step-size methods because of this increased knowledge.

On-Line Adaptive Step-Size Selection

Gradient adaptation has proven to be quite useful for parameter estimation. It can also be used to optimize the step-size parameters in an on-line fashion. This idea has appeared and reappeared in the scientific literature. One of the earliest descriptions of the methodology appears in [10], and it was later reintroduced to both the neural-network [11] and signal-processing [12, 13] communities, where it has become known as the "delta-bar-delta rule" and "gradient step-size method," respectively. An alternative version of

Authorized licensed use limited to: Danmarks Tekniske Informationscenter. Downloaded on April 13,2010 at 12:14:56 UTC from IEEE Xplore. Restrictions apply.